

1 In the claims:

2 1. A virtual input/output (I/O) interconnect mechanism for use in a computer system
3 having a plurality of I/O devices and a plurality of processing units, wherein I/O devices
4 and processing units are coupled by one or more bridge units, the mechanism comprising:

5 an address decode block, comprising:

6 a multiplexer that multiplexes inputs to produce an address, wherein the
7 address relates to a transaction related to a processor unit,

8 a range register decoder that receives the address and provides a
9 destination address of a module to receive the transaction related to the address, and

10 a reroute module identification block that receives the destination address,
11 the reroute module identification block, comprising:

12 an original module identification that provides an address of one or
13 more original modules in the computer system, and

14 a remapped module identification that provides logical destination
15 module identifications of substitute modules in the computer system, wherein a substitute
16 module replaces functions of an original module in the computer system.

17 2. The mechanism of claim 1, wherein the reroute module identification block
18 further comprises a valid bit indication, the valid bit indication indicating when a
19 translation from the original module to the substitute module is valid.

20 3. The mechanism of claim 1, wherein the inputs received by the multiplexer are
21 inputs from the plurality of I/O devices.

22 4. The mechanism of claim 1, wherein the address is an I/O device address.

23 5. The mechanism of claim 1, wherein the destination address is an address of one of
24 the one or more bridge units.

25 6. The mechanism of claim 1, wherein the original module and the substitute module
26 are a same type of device.

27 7. The mechanism of claim 6, wherein the original module and the substitute module
28 are I/O devices.

29 8. The mechanism of claim 7, wherein the I/O devices are ethernet cards.

30 9. The mechanism of claim 1, wherein the substitute module comprises
31 programming to claim transactions addressed to the original module.

32 10. The mechanism of claim 1, wherein a state of the original module is copied to the
33 substitute module.

12. The mechanism of claim 1, wherein the substitute module is an active component of the computer system.

13. A method for rerouting a transaction along a virtual path from an original module in a computer system to a substitute module in the computer system, wherein a transaction is initiated to the original module, and wherein the original module has failed, the method, comprising:

storing a remapped module identification that identifies the substitute module as a substitute for the failed original module;

receiving a transaction for the failed original module, the transaction including an address of the failed original module;

extracting the address;

decoding the address to provide an identification of the failed original module for receipt of the transaction;

comparing the identification of the failed original module to the remapped module identification; and

rerouting the transaction to the substitute module based on the remapped module identification.

14. The method of claim 13, wherein the original module and the substitute module are input/output (I/O) devices.

15. The method of claim 14, wherein the I/O devices are ethernet cards.

16. The method of claim 13, wherein the substitute module is an active component of the computer system.

17. The method of claim 13, wherein the substitute module is an inactive component of the computer system.

18. The method of claim 13, further comprising copying a state of the failed original module to the substitute module.

19. The method of claim 13, further comprising reprogramming the substitute module to claim transactions addressed to the failed original module.

20. The method of claim 13, further comprising setting a valid bit to indicate if a translation from the original module to the substitute module is valid.

21. A method for creating a virtual hardware path in a computer system, comprising:
identifying a failed module in the computer system;

1 identifying a substitute module in the computer system, the substitute module
2 capable of replacing the failed module;
3 storing an identification of the failed module and an identification of the substitute
4 module, wherein the failed module identification is related to the substitute module
5 identification;
6 programming the substitute module to receive transactions addressed to the failed
7 module.

8 22. The method of claim 21, wherein programming the substitute module comprises:
9 copying a state of the failed module to the substitute module; and
10 programming the substitute module to claim the transactions addressed to the
11 failed module.

12 23. An apparatus that creates a virtual hardware path in a computer system,
13 comprising:

14 a reroute module identification block, comprising:
15 an original module identification section, and
16 a substitute module identification section, wherein an identification for a
17 failed original module is entered into a space in the original module identification section
18 and an identification for a substitute module is entered in to a space, corresponding to the
19 space in the original module identification section, in the substitute module section; and
20 programming adapted to the substitute module, wherein the substitute module is
21 reprogrammed to claim transactions addressed to the failed original module and to
22 copying a state of the failed original module to the substitute module.

23 24. A method for substituting operating components for failed components in a
24 computer system, comprising:

25 detecting a failed component;
26 determining if a component of a same type as the failed component exists; and
27 if a substitute component exists:
28 suspending activities to the failed component,
29 copying a state of the failed component to the substitute component,
30 deconfiguring the failed component,
31 updating reroute module identification to remap the failed component to
32 the substitute component,
33 updating configuration registers of the substitute component, and
34 resuming activities to the failed component.

- 1 25. The method of claim 24, wherein a substitute component does not exist, the
- 2 method further comprising invoking an error handler.
- 3 26. The method of claim 24, wherein the activities include direct memory access.